**Software Requirements Specification**

**for**

**<Project>**

**Version 1.0 approved**

**Prepared by <author>**

**<date created>*Software Requirements Specification for <Project> Page ii***

**Table of Contents**

[1. Introduction 3](#_Toc4085)

[1.1 Purpose 3](#_Toc4086)

[1.2 Scope 3](#_Toc4087)

[1.3 Definitions, acronyms, and abbreviations 3](#_Toc4088)

[1.4 References 3](#_Toc4089)

[1.5 Overview 3](#_Toc4090)

[2. Overall Description 4](#_Toc4091)

[2.1 Product Perspective 4](#_Toc4092)

[2.2 Product Functions 4](#_Toc4093)

[3. Specific requirements 5](#_Toc4094)

[3.1 Functional requirements 5](#_Toc4095)

[3.2 Non-functional requirements 5](#_Toc4096)

[3.2.1 Performance Requirements 5](#_Toc4097)

[3.2.2 Other non-functional requirements which are necessary to your system 5](#_Toc4098)

[3.3 Logical database requirements 5](#_Toc4099)

# Introduction

This is a template, you don’t need to fill up every section. Please use it as a reference, and customize it as what you need.

## Purpose

This section should

1. Delineate the purpose of the SRS;
2. Specify the intended audience for the SRS.

## Scope

This section should

1. Identify the software product(s) to be produced by name;
2. Explain what the software product(s) will, and, if necessary, will not do;
3. Describe the application of the software being specified, including relevant benefits, objectives, and goals;
4. Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.

## Definitions, acronyms, and abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

## References

This section should

1. Provide a complete list of all documents referenced elsewhere in the SRS;
2. Identify each document by title, report number (if applicable), date, and publishing organization;
3. Specify the sources from which the references can be obtained.

This information may be provided by reference to an appendix or to another document.

## Overview

This section should

1. Describe what the rest of the SRS contains;
2. Explain how the SRS is organized.

# Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS, and makes them easier to understand.

## Product Perspective

This section should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

## Product Functions

Our (project name) will support four different user types who can interact with the system: Administrator, Patients, Staff and Doctors. All users will need to log into the system before interacting with it, authenticated by providing both a username and password.

Patients may register their own account; however, administrators must verify them before they can be used. Administrators create all Staff and Doctor accounts. Only an administrator can delete accounts.

Once created, users can modify the details of their own accounts. For patients this will include () and for Staff and Doctors this will include ()

Doctors will prescribe exams to patients after they have an appointment, these appointments are not handles by our system. Exam types include () and blood exams can themselves by many types including (). Multiple exams may be prescribed at once.

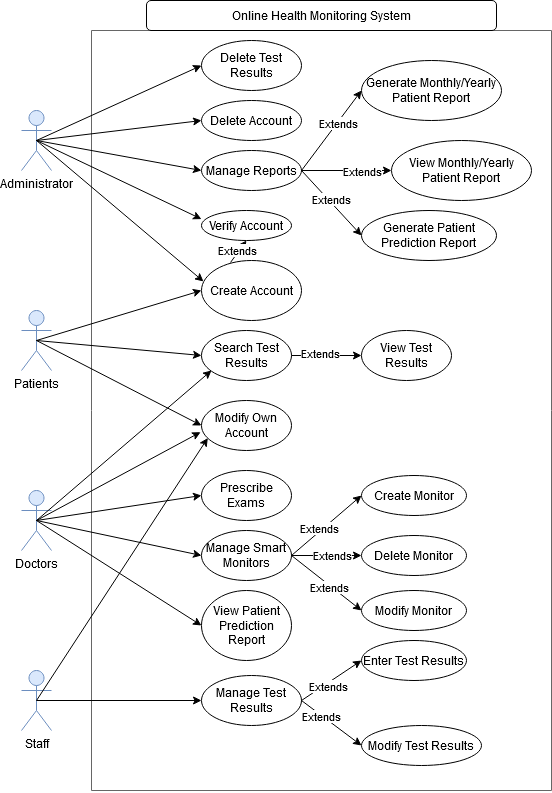
Once an exam is completed staff will input the results into the system. Staff may also alter the results of an exam if needed. Only an administrator may delete exam results entirely.

Doctors and patients may both search and view test results. Patients may only search for their own results, and may search by properties such as (). Doctors can search for results from any patient, and their searches can include properties such as ().

Doctors also have the ability to create, modify and delete smart monitors. The monitors will look for abnormal test results of a given patient as defined by the doctor who created the monitor. Whenever a test result is created or modified, these monitors will check them and if the result is abnormal, will notify the doctor with an email.

Administrators are responsible for generating reports. They may create and view both monthly and yearly reports summarizing all patient health issues and outcomes over the given timeframe. Administrators also create patient prediction reports based on test results in the system, these reports may be viewed by doctors.

The following use-case diagram shows all the main functionalities of the system.

****

# Specific requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

## Functional requirements

This section is important since it lists all functional requirements.

This section should list the functional requirements with details. Please organize the functional requirements into sub-sections with textual description for each requirement.

## Non-functional requirements

This section should list the non-functional requirements with details. Please organize the nonfunctional requirements into sub-sections. All of these requirements should be stated in measurable terms.

### Performance Requirements

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

1. The number of terminals to be supported;
2. The number of simultaneous users to be supported;
3. *Amount and type of information to be handled.*

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms.

For example,

*95% of the transactions shall be processed in less than 1 s.*  rather than,

An operator shall not have to wait for the transaction to complete.

### Other non-functional requirements which are necessary to your system

**……**

## Logical database requirements

This should specify the logical requirements for any information that is to be placed into a database. This may include the following:

1. Types of information used by various functions;
2. Frequency of use;
3. Accessing capabilities;
4. Data entities and their relationships;
5. Integrity constraints;
6. Data retention requirements.